AMENDMENTS TO THE CLAIMS

This listing of claims will replace all prior versions, and listings, of claims in the application.

1. (Previously Presented) An apparatus for use in a computer system comprising:

a memory having stored therein a first packed data comprising at least four data elements and a second packed data comprising at least four data elements; and

a processor coupled to said memory to receive said first packed data and said second packed data, said processor performing operations on data elements in said first packed data and said second packed data to generate a plurality of data elements in a third packed data in response to receiving an instruction, at least two of said plurality of data elements in said third packed data storing the result of multiply-add operations.

27. (Previously Presented) The apparatus of claim 1, said third packed data storing the result of multiply-add operations in a register specified by bits three through five of the instruction.

28. (Previously Presented) The apparatus of claim 27, said processor overwriting said first packed data in the register with said third packed data.

29. (Previously Presented) The apparatus of claim 1, said processor performing operations on four data elements in said first packed data and four data elements in said second packed data.

30. (Previously Presented) The apparatus of claim 1, said processor to generate the plurality of data elements in the third packed data in response to receiving the

instruction, at least four of said plurality of data elements in said third packed data storing the result of multiply-add operations.

31. (Previously Presented) The apparatus of claim 30, said processor performing operations on eight data elements in said first packed data and eight data elements in said second packed data.

32. (Previously Presented) A computer-implemented method responsive to the execution of a single instruction, the computer-implemented method comprising:

A) multiplying together a first value and a second value to generate a first intermediate result;

B) multiplying together a third value and a fourth value to generate a second intermediate result;

C) multiplying together a fifth value and a sixth value to generate a third intermediate result;

D) multiplying together a seventh value and an eighth value to generate a fourth intermediate result;

E) adding together said first intermediate result and said second intermediate result to generate a first data element in a first packed data;

F) adding together said third intermediate result and said fourth intermediate result to generate a second data element in said first packed data;

G) storing said first packed data, and

H) completing execution of said single instruction without adding said first and second data elements.

33. (Currently Amended) The method of claim [[27]] 32 further comprising:

accessing said first, third, fifth, and seventh values from a storage area; and

writing said first packed data over said first, third, fifth, and seventh values in said storage area.

34. (Currently Amended) A processor-implemented method for manipulating a first packed data and a second packed data responsive to the execution of a single instruction, said first packed data including A_1, A_2, A_3, and A_4 as data elements, said second packed data including B_1, B_2, B_3, and B_4 as data elements, said method comprising:

performing the operation $(A_1 \times B_1) + (A_2 \times B_2)$ to generate a first data element in a third packed data;

performing the operation $(A_3 \times B_3) + (A_4 \times B_4)$ to generate a second data element in said third packed data;

storing said third packed data for use as an operand to another instruction following the single instruction.

35. (Previously Presented) The method of claim 34 further comprising:

accessing said first packed data from a register; and

writing said third packed data over said first packed data in said register.

36. (Previously Presented) The method of claim 34, said first packed data also including A_5, A_6, A_7, and A_8 as data elements, said second packed data also including B_5, B_6, B_7, and B_8 as data elements, said method further comprising:

performing the operation $(A_5 \times B_5) + (A_6 \times B_6)$ to generate a third data element in said third packed data;

performing the operation $(A_7 \times B_7) + (A_8 \times B_8)$ to generate a fourth data element in said third packed data.

37. (Currently Amended) A processor-implemented method for manipulating a first packed data and a second packed data responsive to the execution of a single instruction, said first packed data including A_1, A_2, A_3, and A_4 as data elements, said second packed data including B_1, B_2, B_3, and B_4 as data elements, said method comprising:

performing the operation $(A_1 \times B_1) + (A_2 \times B_2)$ responsive to the execution of the single instruction to generate a first data element in a third packed data;

performing the operation $(A_3 \times B_3) + (A_4 \times B_4)$ responsive to the execution of the single instruction to generate a second data element in said third packed data;

storing said third packed data in an architecturally visible register responsive to the execution of the single instruction.

38. (Previously Presented) The method of claim 37, said first packed data also including A_5, A_6, A_7, and A_8 as data elements, said second packed data also including B_5, B_6, B_7, and B_8 as data elements, said method further comprising:

performing the operation $(A_5 \times B_5) + (A_6 \times B_6)$ to generate a third data element in said third packed data;

performing the operation $(A_7 \times B_7) + (A_8 \times B_8)$ to generate a fourth data element in said third packed data.

39. (Previously Presented) The method of claim 38 further comprising:

accessing said first packed data from the architecturally visible register; and

storing said third packed data over said first packed data in the architecturally visible register.

40. (Currently Amended) In a computer system having stored therein a first packed data and a second packed data each containing initial data elements, each of said initial data elements in said first packed data having a corresponding initial data element in said second packed data, a method for performing multiply add operations in response to a single instruction, said method comprising:

multiplying together said corresponding initial data elements in said first packed data and said second packed data to generate corresponding intermediate data elements, said intermediate data elements being divided into a number of sets;

generating a plurality of result data elements, a first of said plurality of result data elements [[representing]] including the sum of said intermediate [[result]] data elements in a first of said number of sets, a second of said plurality of result data elements [[representing]] including the sum of said intermediate [[result]] data elements in a second of said number of sets; and

completing execution of said single instruction without summing said plurality of result data elements.

41. (Previously Presented) The method of claim 40 further comprising:

storing said plurality of result data elements as a third packed data for use as an operand to another instruction.

42. (Previously Presented) The method of claim 40 further comprising:

accessing said first and second packed data from a register; and

writing said plurality of result data elements over said first packed data in said register.

43. (Previously Presented) The method of claim 40 comprising:

completing execution of said single instruction without summing said plurality of result data elements in response to said single instruction.

44. (Currently Amended) The method of claim 40 comprising:

generating a third of said plurality of result data elements [[representing]] including the sum of said intermediate result data elements in a third of said number of sets, and a fourth of said plurality of result data elements [[representing]] including the sum of said intermediate result data elements in a fourth of said number of sets; and

completing execution of said single instruction without summing said plurality of result data elements in response to said single instruction.

45. (Currently Amended) A processor-implemented method for manipulating a first packed data and a second packed data responsive to the execution of a single instruction, said first packed data including A_1, A_2, A_3, and A_4 as data elements, said second packed data including B_1, B_2, B_3, and B_4 as data elements, said method comprising:

multiplying together A1 and B1 to generate a first intermediate result;

multiplying together A2 and B2 to generate a second intermediate result;

multiplying together A3 and B3 to generate a third intermediate result; and

multiplying together A4 and B4 to generate a fourth intermediate result;

performing in parallel the following:

adding together said first intermediate result and said second intermediate result to generate a first data element in a third packed data; and

adding together said third intermediate result and said fourth intermediate result to generate a second data element in said third packed data; and

saving said third packed data for use as an operand to another instruction.

46. (Previously Presented) The method of claim 45 further comprising:

accessing said first and second packed data from a storage area; and

writing said third packed data in said storage area.

47. (Previously Presented) The method of claim 45 said first packed data also including A_5, A_6, A_7, and A_8 as data elements, said second packed data also including B_5, B_6, B_7, and B_8 as data elements, said method further comprising:

multiplying together A5 and B5 to generate a fifth intermediate result;

multiplying together A6 and B6 to generate a sixth intermediate result;

multiplying together A7 and B7 to generate a seventh intermediate result; and

multiplying together A8 and B8 to generate an eighth intermediate result;

performing in parallel the following:

adding together said fifth intermediate result and said sixth intermediate result to generate a third data element in said third packed data; and

adding together said seventh intermediate result and said eighth intermediate result to generate a fourth data element in said third packed data accessing said first packed data from a storage area; and

overwriting said first packed data with said third packed data in said storage area.

48. (Previously Presented) An apparatus comprising:

a first storage area to store packed data;

a second storage area to store packed data;

an execution unit, responsive to a first instruction specifying a first source indicating the first storage area and a second source indicating the second storage area, to sum a first product of multiplication of a first element stored in the first storage area by a second element stored in the second storage area, with a second product of multiplication of a third element stored in the first storage area by a fourth element stored in the second storage area, producing a first result, the execution unit, further responsive to the first instruction, to sum a third product of multiplication of a fifth element stored in the first storage area by a sixth element stored in the second storage area, with a fourth product of multiplication of a seventh element stored in the first storage area by an eighth element stored in the second storage area, producing a second result, and the execution unit to store both first and second results into a destination storage area specified by the first instruction.

49. (Previously Presented) The apparatus of claim 48 wherein the destination specified by the first instruction is the same as the first source.

50. (Previously Presented) The apparatus of claim 49 wherein the destination and the first source are specified by bits three through five of the first instruction.

51. (Previously Presented) The apparatus of claim 50 wherein the second source is specified by bits zero through two of the first instruction.

52. (Previously Presented) The apparatus of claim 48 wherein the second source specified by the first instruction indicates that the second storage area is a memory location in an addressable memory space.

53. (Previously Presented) The apparatus of claim 48, said execution unit to access the first and second results stored in the destination storage area as a third source specified by a second instruction.

54. (Previously Presented) The apparatus of claim 48 further comprising a decoder to decode a first set of instructions of one or more instruction formats having a first field and a second field, the first set of instructions including said first instruction with the first field specifying the first source and the second field specifying the second source.

55. (Previously Presented) The apparatus of claim 54 wherein said first instruction of the one or more instruction formats includes a 24-bit instruction format.

56. (Previously Presented) The apparatus of claim 54 wherein said first instruction of the one or more instruction formats includes a 32-bit instruction format.

57. (Previously Presented) The apparatus of claim 54, said execution unit, further responsive to the first instruction, to sum a fifth product of multiplication of a

ninth element stored in the first storage area by a tenth element stored in the second storage area, with a sixth product of multiplication of an eleventh element stored in the first storage area by a twelfth element stored in the second storage area, producing a third result, and to sum a seventh product of multiplication of a thirteenth element stored in the first storage area by a fourteenth element stored in the second storage area, with an eighth product of multiplication of a fifteenth element stored in the first storage area by a sixteenth element stored in the second storage area, producing a fourth result, and finally to store the first, second, third and fourth results into the destination storage area specified by the first instruction.

58. (Previously Presented) The apparatus of claim 54 wherein the one or more instruction formats correspond with the integer opcode format of the PENTIUM processor family.

59. (Previously Presented) The apparatus of claim 58 wherein the decoder comprises a table of the first set of instructions with which to look up said first instruction.

60. (Previously Presented) The apparatus of claim 58 wherein the decoder comprises integrated hardware.

61. (Previously Presented) The apparatus of claim 58 wherein the decoder comprises a combination of integrated hardware and a table of the first set of instructions with which the integrated hardware can look up said first instruction.

62. (Previously Presented) An apparatus comprising:

a first storage area to store packed data, the first storage area corresponding to a first source;

a second storage area to store packed data, the second storage area corresponding to a second source;

a decoder to decode a first set of one or more instruction formats having a first field to specify the first source and a second field to specify the second source;

an execution unit, responsive to the decoder decoding a first instruction of the first set of one or more instruction formats, to sum a first product of multiplication of a first element stored in the first storage area by a second element stored in the second storage area, with a second product of multiplication of a third element stored in the first storage area by a fourth element stored in the second storage area, producing a first result;

the execution unit, further responsive to the decoding of the first instruction, to sum a third product of multiplication of a fifth element stored in the first storage area by a sixth element stored in the second storage area, with a fourth product of multiplication of a seventh element stored in the first storage area by an eighth element stored in the second storage area, producing a second result; and

a third storage area to store packed data, the third storage area corresponding to a destination specified by the first instruction to store the first and second results.

63. (Previously Presented) The apparatus of claim 62 wherein said first instruction of the first set of one or more instruction formats has a 24-bit instruction format.

64. (Previously Presented) The apparatus of claim 62 wherein said first instruction of the first set of one or more instruction formats has a 32-bit instruction format.

65. (Previously Presented) The apparatus of claim 64, the execution unit, further responsive to the decoder decoding the first instruction, to sum a fifth product of multiplication of a ninth element stored in the first storage area by a tenth element

stored in the second storage area, with a sixth product of multiplication of an eleventh element stored in the first storage area by a twelfth element stored in the second storage area, producing a third result;

the execution unit, further responsive to the decoder decoding the first instruction, to sum a seventh product of multiplication of a thirteenth element stored in the first storage area by a fourteenth element stored in the second storage area, with an eighth product of multiplication of a fifteenth element stored in the first storage area by a sixteenth element stored in the second storage area, producing a fourth result; and

the third storage area corresponding to the destination specified by the first instruction to store the first, second, third and fourth results.

66. (Previously Presented) The apparatus of claim 62 wherein the destination is specified by the first field.

67. (Previously Presented) The apparatus of claim 66 wherein the first field corresponds to bits 3-5 of the first instruction.

68. (Previously Presented) The apparatus of claim 62 wherein the destination is specified by the second field.

69. (Previously Presented) The apparatus of claim 68 wherein the second field corresponds to bits 0-2 of the first instruction.

70. (Previously Presented) The apparatus of claim 62 wherein the first set of one or more instruction formats corresponds with the set of integer opcode formats of the PENTIUM processor family.

71. (Previously Presented) The apparatus of claim 70 wherein the decoder comprises a look-up table.

72. (Previously Presented) The apparatus of claim 70 wherein the decoder comprises integrated hardware.

73. (Previously Presented) The apparatus of claim 70 wherein the decoder comprises a combination of a look-up table and integrated hardware.

74. (Previously Presented) The apparatus of claim 70 wherein the execution unit comprises a mechanism to initiate a series of operations including multiplications and sums.

75. (Previously Presented) The apparatus of claim 70 wherein the execution unit comprises integrated hardware.

76. (Previously Presented) The apparatus of claim 70 wherein the execution unit comprises a combination of integrated hardware and a mechanism to initiate a series of operations executable by the integrated hardware.

77. (Previously Presented) The apparatus of claim 62, wherein the first through the eighth elements are signed 16-bit word elements.

78. (Previously Presented) The apparatus of claim 62, wherein the first, second and third storage areas are to store 64-bit packed data.

79. (Previously Presented) The apparatus of claim 78, wherein the first result corresponds to bits 0-31 of the destination and the second result corresponds to bits 32-63 of the destination.

80. (Previously Presented) The apparatus of claim 79, wherein the first and second elements corresponds to bits 0-15, the third and fourth elements corresponds to bits 16-31, the fifth and sixth elements corresponds to bits 32-47, and the seventh and eighth elements corresponds to bits 48-63 of the first and second sources respectively.

81. (Previously Presented) The apparatus of claim 78, wherein the first, second and third storage areas are to also store 80-bit floating-point data.

82. (Previously Presented) A processor comprising:

a decoder to decode an instruction having a first format, said first format operable to identify a first set of packed data including four elements A_1, A_2, A_3 and A_4, and to identify a second set of packed data including four elements B_1, B_2, B_3 and B_4, said decode unit to initiate a first set of operations on the first and second sets of packed data responsive to decoding the instruction; and

an execution unit to perform a first operation of the first set of operations initiated by the decoder to produce a first result corresponding to the sum of products, $(A_1 \times B_1) + (A_2 \times B_2)$, and to perform a second operation of the first set of operations initiated by the decoder to produce a second result corresponding to the sum of products, $(A_3 \times B_3) + (A_4 \times B_4)$.

83. (Previously Presented) The apparatus of claim 82, the first set of packed data also including four more elements A_5, A_6, A_7 and A_8, and the second set of packed data also including four more elements B_5, B_6, B_7 and B_8; and the execution unit to perform a third operation of the first set of operations initiated by the decoder to produce a third result corresponding to the sum of products, $(A_5 \times B_5) + (A_6 \times B_6)$, and to perform a fourth operation of the first set of operations initiated

by the decode unit to produce a fourth result corresponding to the sum of products, $(A_7 \times B_7) + (A_8 \times B_8)$.

84. (Previously Presented) The apparatus of claim 83, the execution unit to perform a fifth operation of the first set of operations initiated by the decoder to store the first, second, third and fourth results in an architecturally visible register.

85. (Previously Presented) The apparatus of claim 84 wherein the execution unit accesses the first set of packed data from the architecturally visible register and overwrites the first set of packed data in the architecturally visible register with the first, second, third and fourth results.

86. (Previously Presented) The apparatus of claim 85 wherein said instruction has a 32-bit format.

87. (Previously Presented) The apparatus of claim 82, the execution unit to perform a third operation of the first set of operations initiated by the decoder to store the first and second results in an architecturally visible register.

88. (Previously Presented) The apparatus of claim 87 wherein the execution unit accesses the first set of packed data from the architecturally visible register and overwrites the first set of packed data in the architecturally visible register with the first and second results.

89. (Previously Presented) The apparatus of claim 88 wherein the instruction has a 24-bit format.

90. (Previously Presented) The apparatus of claim 85 or of claim 88 wherein the second set of packed data is identified by the first format of the instruction as a memory location in an addressable memory space of the processor.

91. (Currently Amended) An apparatus comprising:

a decoder to decode an instruction having a first format, said first format operable to identify a first M×N-bit source storing a first plurality of M packed N-bit data elements, and to identify a second M×N-bit source storing a second plurality of M packed N-bit data elements, each element of the second plurality corresponding to an element of the first plurality, said decoder to initiate one or more operations on the first and second pluralities of packed N-bit data elements responsive to decoding the instruction, wherein M is an even number; and

an execution unit to perform the one or more operations initiated by the decoder to produce a third plurality of M/2 packed 2N-bit results corresponding to M/2 sums of [[paired]] grouped products of corresponding elements multiplied from the first and second pluralities, said first format also operable to identify a M×N-bit destination for the execution unit to store the third plurality in of M/2 packed 2N-bit results.

92. (Previously Presented) The apparatus of claim 91, wherein M is equal to four.

93. (Previously Presented) The apparatus of claim 92, wherein the packed N-bit data elements of the first and second pluralities are signed 16-bit word elements.

94. (Currently Amended) The apparatus of claim 93, wherein the M/2 sums of [[paired]] grouped products are truncated 32-bit doubleword elements.

95. (Previously Presented) The apparatus of claim 91, wherein M is equal to eight.

96. (Previously Presented) The apparatus of claim 95 wherein the execution unit accesses the first plurality of M packed N-bit data from an architecturally visible register and overwrites the first plurality of M packed N-bit data in the

architecturally visible register with the third plurality of M/2 packed 2N-bit results.

97. (Previously Presented) The apparatus of claim 96 wherein the instruction has a 32-bit first format.

98. (Previously Presented) The apparatus of claim 96 wherein the second plurality of M packed N-bit data is identified by the first format of the instruction as a memory location in an addressable memory space.

99. (Previously Presented) The apparatus of claim 91, wherein a plurality of the one or more operations performed by the execution unit are performed in parallel.

100. (Previously Presented) The apparatus of claim 91, wherein a plurality of the one or more operations performed by the execution unit are performed in series.

101. (Previously Presented) An apparatus comprising:

means for decoding an instruction to identify a first set of M packed N-bit data elements including A_1, A_2, A_3 and A_4, a second set of M packed N-bit data elements including B_1, B_2, B_3 and B_4, and an M×N-bit destination;

means for initiating a first set of operations on elements of the first and second sets of M packed N-bit data elements;

means for executing a first operation of the first set of operations to produce a first result substantially equal to $(A_1 \times B_1) + (A_2 \times B_2)$;

means for executing a second operation of the first set of operations to produce a second result substantially equal to $(A_3 \times B_3) + (A_4 \times B_4)$; and

means for storing the first and second results as packed 2N-bit data elements at the identified M×N-bit destination.

102. (Previously Presented) The apparatus of claim 101, wherein M is equal to four.

103. (Previously Presented) The apparatus of claim 101, wherein M is equal to eight.

104. (Previously Presented) The apparatus of claim 103, said first set of M packed N-bit data elements further including A_5, A_6, A_7 and A_8, and said second first set of M packed N-bit data elements further including B_5, B_6, B_7 and B_8, the apparatus further comprising:

means for executing a third operation of the first set of operations to produce a third result substantially equal to $(A_5 \times B_5) + (A_6 \times B_6)$;

means for executing a fourth operation of the first set of operations to produce a fourth result substantially equal to $(A_7 \times B_7) + (A_8 \times B_8)$; and

means for storing the first, second, third and fourth results as packed 2N-bit data elements at the identified M×N-bit destination.

105. (Previously Presented) The apparatus of claim 104 wherein the means for executing the first, second, third and fourth operations access the first set of M packed N-bit data from an architecturally visible register and overwrites the first set of M packed N-bit data in the architecturally visible register with the first, second, third and fourth results.

106. (Previously Presented) The apparatus of claim 101, wherein the first and second results are truncated 32-bit doubleword elements.

107. (Previously Presented) The apparatus of claim 106, wherein the packed N-bit data elements of the first and second sets are signed 16-bit word elements.

108. (Previously Presented) The apparatus of claim 101 wherein the means for executing the first and second operations access the first set of M packed N-bit data from an architecturally visible register and overwrites the first set of M packed N-bit data in the architecturally visible register with the first and second results.

109. (Previously Presented) An article of manufacture comprising:

a machine-accessible medium including data that, when accessed by a machine, cause the machine to:

initiate, responsive to a single instruction, a first set of operations on elements of a first M×N-bit packed data source including N-bit elements A_1, A_2, A_3 and A_4, and a second M×N-bit packed data source including N-bit elements B_1, B_2, B_3 and B_4;

execute a first operation of the first set of operations in producing a first result substantially equal to $(A_1 \times B_1) + (A_2 \times B_2)$;

execute a second operation of the first set of operations in producing a second result substantially equal to $(A_3 \times B_3) + (A_4 \times B_4)$; and

store the first and second results as packed 2N-bit data elements of an M×N-bit packed data destination.

110. (Previously Presented) The article of manufacture of claim 109, wherein M is equal to four.

111. (Previously Presented) The article of manufacture of claim 109, wherein M is equal to eight.

112. (Previously Presented) The article of manufacture of claim 111, the machine-accessible medium further including data that, when accessed by a machine, cause the machine to:

initiate the first set of operations on elements of said first M×N-bit packed data source further including N-bit elements A_5, A_6, A_7 and A_8, and said second M×N-bit packed data source further including N-bit elements B_5, B_6, B_7 and B_8;

execute a third operation of the first set of operations in producing a third result substantially equal to $(A_5 \times B_5) + (A_6 \times B_6)$;

execute a fourth operation of the first set of operations in producing a fourth result substantially equal to $(A_7 \times B_7) + (A_8 \times B_8)$; and

store the first, second, third and fourth results as packed 2N-bit data elements of an M×N-bit packed data destination.

113. (Previously Presented) The article of manufacture of claim 112, the machine accessible medium further including data that, when accessed by a machine, cause the machine to:

access the first set of M packed N-bit data from a register; and

overwrite the first set of M packed N-bit data in the register with the first, second, third and fourth results.

114. (Previously Presented) The article of manufacture of claim 109, wherein the first and second results are truncated 32-bit doubleword elements.

115. (Previously Presented) The article of manufacture of claim 114, the machine-accessible medium further including data that, when accessed by a machine, cause the machine to:

access the first set of M packed N-bit data from a register; and

overwrite the first set of M packed N-bit data in the register with the first and second results.

116. (Currently Amended) A computing system comprising:

a memory to store a first M×N-bit data consisting of a first plurality of M packed N-bit data elements, and a second M×N-bit data consisting of a second plurality of M packed N-bit data elements, wherein M is an even number; and

a processor operatively coupled with the memory to produce a third plurality of M/2 packed 2N-bit results in response to an instruction having a first format, said first format operable to identify a first source corresponding to the first M×N-bit data and a second source corresponding to the second M×N-bit data, the third plurality of M/2 packed 2N-bit results [[representing]] including M/2 sums of paired products of corresponding data elements multiplied from the first and second pluralities of M packed N-bit data elements.

117. (Previously Presented) The computing system of Claim 116 further comprising:

a digital audio device operatively coupled with the memory to provide one or more N-bit data element of the first M×N-bit data; and

an optical disk drive operatively coupled with the processor to receive an optical disk for storing audio data.

118. (Previously Presented) The computing system of Claim 116 further comprising:

a digital motion video device operatively coupled with the memory to provide one or more N-bit data element of the first M×N-bit data; and

an optical disk drive operatively coupled with the processor to receive an optical disk for storing digital motion video data.

119. (Previously Presented) The computing system of Claim 116 further comprising:

a digital graphics device operatively coupled with the memory to provide one or more N-bit data element of the first M×N-bit data; and

an optical disk drive operatively coupled with the processor to receive an optical disk for storing digital still image data.

120. (Currently Amended) A computing system adapted to process digital motion video signals, the system comprising:

a memory to store a first M×N-bit data consisting of a first plurality of M packed N-bit data elements, and a second M×N-bit data consisting of a second plurality of M packed N-bit data elements, where M is an even number;

a processor operatively coupled with the memory to produce a third plurality of M/2 packed 2N-bit results in response to an instruction having a first format, said first format operable to identify a first source corresponding to the first M×N-bit data and a second source corresponding to the second M×N-bit data, the third plurality of M/2 packed 2N-bit results [[representing]] including M/2 sums of [[paired]] grouped products of corresponding data elements multiplied from the first and second pluralities of M packed N-bit data elements; and

a digital motion video device operatively coupled with the memory to provide one or more N-bit data element of the first M×N-bit data.

121. (Currently Amended) A computing system adapted to process digital still images, the system comprising:

a memory to store a first M×N-bit data consisting of a first plurality of M packed N-bit data elements, and a second M×N-bit data consisting of a second plurality of M packed N-bit data elements, where M is an even number;

a processor operatively coupled with the memory to produce a third plurality of M/2 packed 2N-bit results in response to an instruction having a first format, said first format operable to identify a first source corresponding to the first M×N-bit data and a second source corresponding to the second M×N-bit data, the third plurality of M/2 packed 2N-bit results [[representing]] including M/2 sums of paired products of corresponding data elements multiplied from the first and second pluralities of M packed N-bit data elements; and

a digital graphics device operatively coupled with the memory to provide one or more N-bit data element of the first M×N-bit data.

122. (Currently Amended) A computing system adapted to process audio signals, the system comprising:

a memory to store a first M×N-bit data consisting of a first even numbered plurality of M packed N-bit data elements, and a second M×N-bit data consisting of a second even numbered plurality of M packed N-bit data elements;

a processor operatively coupled with the memory to produce a third plurality of M/2 packed 2N-bit results in response to an instruction having a first format, said

first format operable to identify a first source corresponding to the first M×N-bit data and a second source corresponding to the second M×N-bit data, the third plurality of M/2 packed 2N-bit results representing M/2 sums of paired products of corresponding data elements multiplied from the first and second pluralities of M packed N-bit data elements; and

a digital audio device operatively coupled with the memory to provide one or more N-bit data element of the first M×N-bit data.

123. (Previously Presented) The apparatus of claim 122 wherein the first source is identified by the first format of the instruction as a register to store a copy of the first M×N-bit data.

124. (Previously Presented) The apparatus of claim 123 wherein the second source is identified by the first format of the instruction as a register to store a copy of the second M×N-bit data.

125. (Previously Presented) The apparatus of claim 123 wherein the second source is identified by the first format of the instruction as a location in an addressable memory space of the processor storing the second M×N-bit data.

126. (Previously Presented) The computing system of claim 124 wherein the instruction has a 24-bit first format.

127. (Previously Presented) The computing system of claim 124 wherein the instruction has a 32-bit first format.

128. (New) The method of claim 40, wherein the first of said plurality of result data elements represents the sum of said intermediate result data elements in a first of said number of sets, and wherein the second of said plurality of result data

elements represents the sum of said intermediate result data elements in a second of said number of sets.

129. (New) The apparatus of claim 91, wherein the third plurality of M/2 packed 2N-bit results correspond to M/2 sums of paired products of corresponding elements multiplied from the first and second pluralities.

130. (New) The computing system of claim 116, wherein the third plurality of M/2 packed 2N-bit results represent M/2 sums of paired products of corresponding data elements multiplied from the first and second pluralities of M packed N-bit data elements.

131. (New) The computing system of claim 120, wherein the third plurality of M/2 packed 2N-bit results represent M/2 sums of paired products of corresponding data elements multiplied from the first and second pluralities of M packed N-bit data elements.

132. (New) The computing system of claim 121, wherein the third plurality of M/2 packed 2N-bit results represent M/2 sums of paired products of corresponding data elements multiplied from the first and second pluralities of M packed N-bit data elements.

133. (New) An apparatus comprising:

a memory to store a first packed data including at least four data elements and a second packed data including at least four data elements; and

a processor coupled with the memory to receive the first packed data and the second packed data, the processor to perform operations on data elements in the first packed data and the second packed data to generate a plurality of data

elements in a third packed data in response to receiving an instruction, at least two of the plurality of data elements in the third packed data resulting from multiply-add operations.

134. (New) The apparatus of claim 133, wherein the processor is to overwrite the first packed data with the third packed data.

135. (New) The apparatus of claim 133, wherein at least four data elements of the third packed data result from multiply-add operations.